

Projeto de Programação (T1) - Versão para Equipe de 3 participantes

O tema do projeto é a geração automática de exercícios a partir de uma base de dados organizada como arquivos-texto. Há 3 tipos de exercícios: "múltipla escolha", "verdadeiro ou falso" e "preencha as lacunas". O programa deve partir de um arquivo que especifica quantas questões devem ser geradas, quais os assuntos, os níveis de dificuldade e os tipos das questões. O programa cria o arquivo de questões sorteando aleatoriamente, a partir da base de dados de fatos sobre o assunto, questões que atendam à especificação.

Exemplos de exercícios dos 3 tipos podem ser encontrados no final de qualquer capítulo de DEITEL e DEITEL (2011)¹. Confira, por exemplo, os capítulos 7 (Ponteiros em C) e 11 (Processamento de Arquivos em C). As questões abaixo são adaptadas do livro:

Múltipla escolha:

Assinale a alternativa FALSA:

- () A função `fopen` retorna `NULL` se não conseguir abrir um arquivo.
- () Para gravar vários elementos de um array, forneça na chamada de `fwrite` um ponteiro para um array como argumento, e o número de elementos a serem gravados como o terceiro argumento.
- () A função `fseek` só pode fazer uma busca a partir do início de um arquivo.

Verdadeiro ou falso (V/F):

Indique se as seguintes sentenças são falsas ou verdadeiras. Se a resposta for falsa, explique.

- a) O operador de endereço (&) pode ser aplicado somente a constantes, a expressões e a variáveis declaradas com a classe de armazenamento `register`.
- b) Um ponteiro que é declarado como `void` pode ser desreferenciado.
- c) Ponteiros de diferentes tipos podem ser atribuídos uns aos outros sem uma operação de coerção (`cast`).

Preencha as lacunas:

Preencha os espaços em cada uma das sentenças:

- a) Uma variável de ponteiros contém como valor o (a) _____ de outra variável.
- b) Os três valores que podem ser usados para inicializar um ponteiro são _____, _____ e _____.
- c) O único inteiro que pode ser atribuído a um ponteiro é _____.

ROTEIRO

I. Definição do formato da base de dados de conteúdo e inserção de sentenças

A primeira tarefa é definir o formato da sentenças armazenadas nos arquivos e criar um programa que minimamente recebe, do usuário, novas sentenças e suas opções e as grava no final de arquivos indicados pelo usuário.

1. A definição da representação das sentenças nos arquivos espelha a definição de um registro (isto é, uma *struct*). Existem dois tipos de sentenças, e, portanto, dois tipos de registros:
 - Sentenças V/F, que servem tanto para geração de questões de "Múltipla escolha" quanto "V/F";
 - Sentenças com lacunas, que servem para geração de questões "Preencha as lacunas".
- a. Os dois tipos de registros têm em comum que representam:
 - Assunto principal (por exemplo, "Arquivos")
 - Assunto secundário (por exemplo, "Arquivos binários")
 - Nível de dificuldade (espaço para projeto: faixa de valores, possibilidade de ser um campo em branco)
 - [Opcional no projeto] Peso (espaço para projeto: faixa de valores, possibilidade de ser um campo em branco), a probabilidade de sorteio da sentença, neste caso, é proporcional ao peso em comparação às outras sentenças.
- b. O registro do tipo V/F possui uma formulação correta e outra formulação que é contrária (e, portanto, falsa). Por exemplo, "Em C, só há passagem de parâmetros por valor." e "Em C, há passagem de valor por referência.", respectivamente. Ou: "O operador & não pode aparecer no lado esquerdo de uma atribuição." e "O operador & pode aparecer no lado esquerdo de uma atribuição."
- c. O registro do tipo "lacunas" possui:
 - A quantidade de lacunas na sentença
 - A formulação da sentença com lacunas (por exemplo, "C foi criada por _____.")
 - Os termos que completam corretamente a sentença (no exemplo acima, "Dennis Ritchie")
2. O programa para inserção de sentenças em arquivos recebe:
 - O nome do arquivo (espaço para projeto: uma possibilidade é receber essa informação na invocação do programa)

¹ DEITEL, PAUL J.; DEITEL, HARVEY M. C : **como programar**. 6. ed., atual. -. São Paulo: Pearson Prentice Hall, 2011. (Disponível na Biblioteca Virtual da Pearson)

- Todas as informações de 1 ou mais sentenças, e grava tudo isso no final do arquivo.
3. Criar arquivos de teste retirando exemplos de DEITEL e DEITEL (2011).

II. Geração de questões

1. O conteúdo que alimenta o programa pode estar em diversos arquivos. O programa pode abrir mais de um arquivo ao mesmo tempo para geração das questões.
 - a. Cada arquivo pode conter sentenças de qualquer tipo, assunto ou nível de dificuldade.
 - b. O usuário deve poder informar os arquivos (espaço para projeto: uma opção é fazer isso na invocação do programa).
2. A especificação das questões é feita por um único arquivo.
 - a. O arquivo pode conter a quantidade de questões (alternativamente, o programa pode calcular percorrendo o arquivo).
 - b. Para cada questão a ser gerada, o arquivo deve conter minimamente o assunto e o tipo de questão (múltipla escolha, V/F ou lacunas). Além disso, podem estar presentes outros atributos (v. I.1.a, I.1.b, I.1.c).
3. Para gerar questões de múltipla escolha:
 - Sortear a quantidade de **opções** entre 3 e 6, com maior chances para 5
 - Sortear se o **enunciado** pede a opção CORRETA ou a opção FALSA
 - Sortear qual das opções será a **resposta**
 - Sortear entre as sentenças de mesmo assunto e nível de dificuldade disponíveis nos arquivos, e usar as formulações correta e falsa de acordo com o enunciado e a resposta
4. Para gerar questões de V/F:
 - Sortear a quantidade de itens entre 3 e 8
 - Sortear se a resposta de cada item é V ou F
 - Verificar se os itens são todos V ou se todos são F; em caso afirmativo, refazer o sorteio
 - Sortear entre as sentenças de mesmo assunto e nível de dificuldade disponíveis nos arquivos, e usar a formulação correta para itens V e contrária, para F
5. Para gerar questões de "preencher lacunas"
 - Sortear a quantidade de itens entre 2 e 5
 - Sortear entre as sentenças de mesmo assunto e nível de dificuldade disponíveis nos arquivos
6. As questões devem ser numeradas em ordem crescente a começar de 1, e possuir um enunciado padrão conforme o tipo.

III. Funcionalidades adicionais

Escolha um ou dois itens abaixo para implementar. Conforme a dificuldade do item escolhido e a qualidade da implementação, a pontuação referente a esta parte pode ser total. Se o grupo optar por não fazer essa parte, isso é absolutamente compreensível e não frustra as expectativas com respeito ao projeto apresentado, apenas reduz a pontuação máxima possível (confira na pontuação).

1. Consulta e edição dos arquivos de conteúdo. O programa especificado em I.2 pode ser expandido, permitindo-se a consulta e edição das sentenças ali existentes.
2. Sorteio guiado. O programa especificado em II pode ter uma interface de acompanhamento do sorteio, em que o usuário pode confirmar uma opção sorteada, ou selecionar manualmente por outras alternativas.
3. Geração de gabarito. Um arquivo adicional contendo as respostas é gerado. Nesse caso, é preciso identificar o arquivo de questões e o arquivo de respostas com um mesmo número. Esse número pode estar no conteúdo do arquivo, em seu nome ou em ambos.
4. Outras ideias de melhoria da interface com usuário.

PONTUAÇÃO

Objetivo	Pontuação máxima
Sentenças e questões de um dos dois tipos: múltipla escolha ou V/F	5,0
Sentenças e questões do outro tipo	1,5
Sentenças e questões de lacunas	1,5
Funcionalidades adicionais	2,0

ENTREGA

Arquivo comprimido (formato ZIP) com códigos-fontes dos programas e arquivos usados no teste, no Google Classroom.